上海交通大學
SHANGHAI JIAO TONG UNIVERSITY

IVM
Image, Video, and Multimedia Communications Laboratory

Digital Image Processing

Hongkai Xiong
熊红凯
http://ivm.sjtu.edu.cn

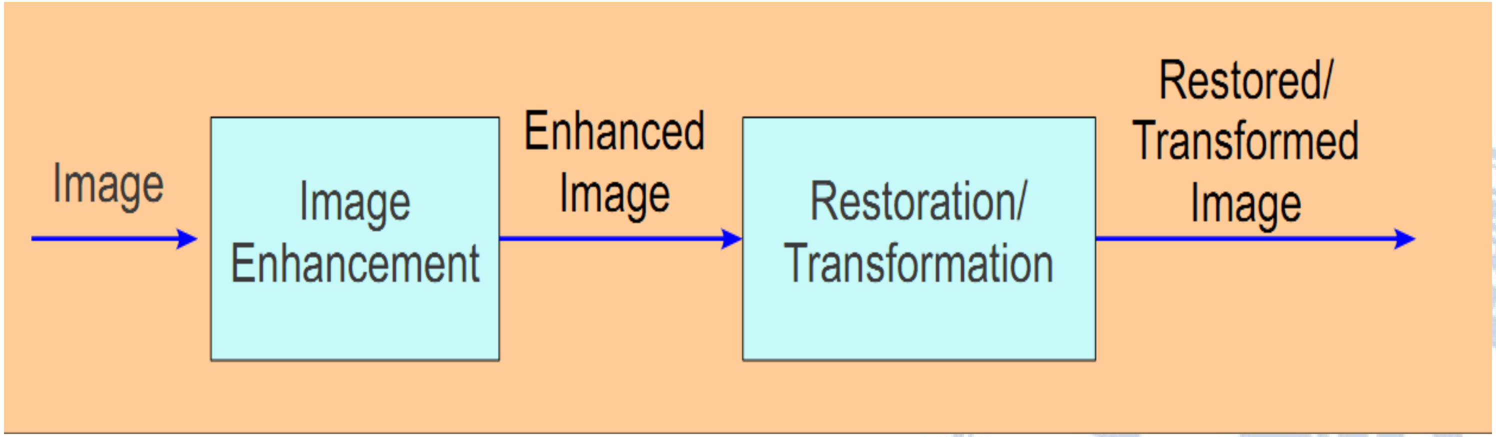
电子工程系
上海交通大学
2016

This lecture will cover

- Representation and Descriptor
  - SIFT descriptor
  - Boundary descriptor
  - Region descriptor

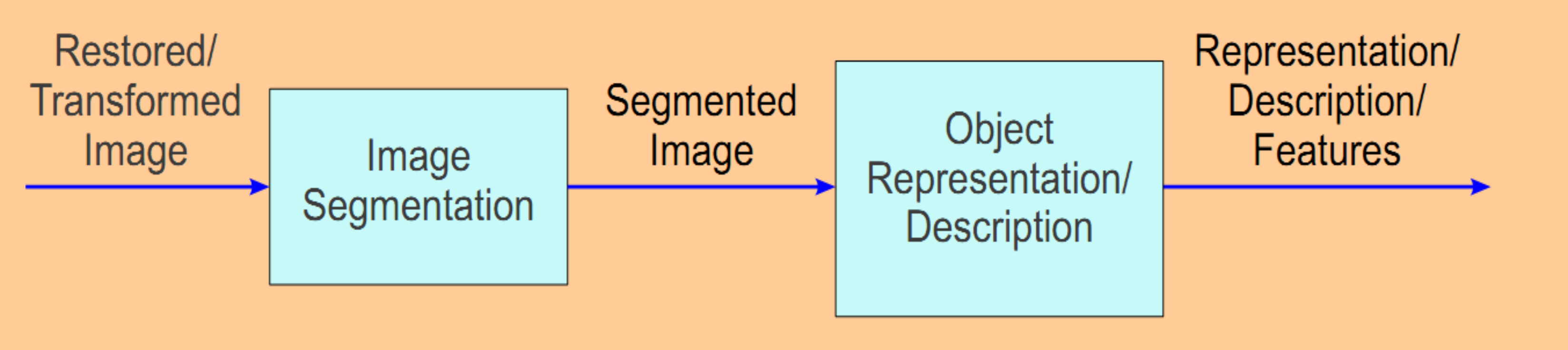
Low-level image processing

- Image enhancement, restoration, transformation…



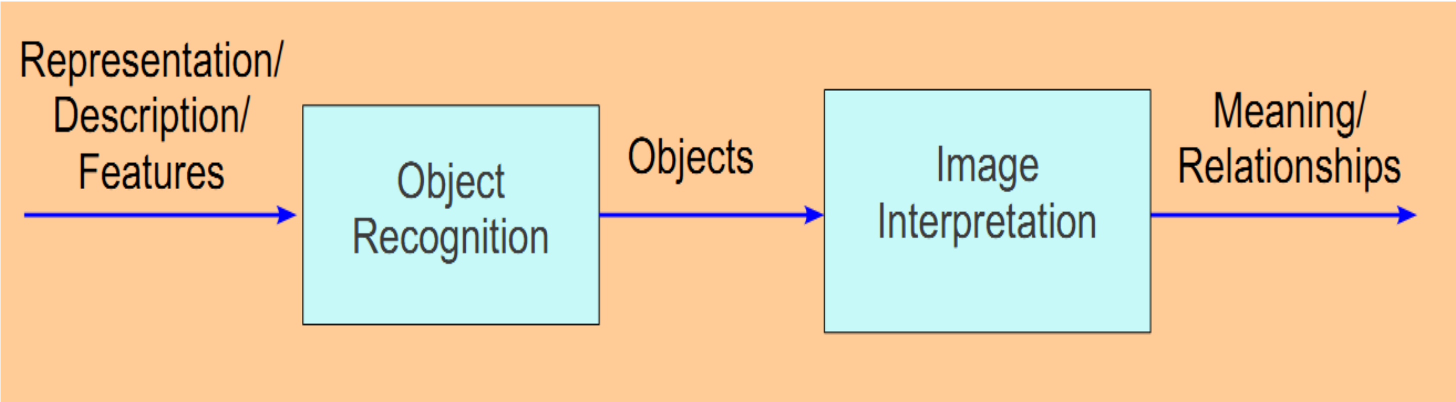
Mid-level image processing (image understanding)

- Object representation, description



High-level image processing (recognition and interpretation)

- Object recognition, interpretation of object relationships



Goal

- To represent and describe the resulting aggregate of segmented pixels in a form suitable for further computer processing.
- There are two information for representing a region:

I. external characteristic (e.g., shape, structure relation)
II. internal characteristic (e.g., gray level, color, texture)

Continued

- In general, description is usually based on the representation.
- It is important that the features selected as descriptors be as insensitive as possible to such variations as changes in size, translation, and rotation

Descriptors – SIFT (A brief introduction)

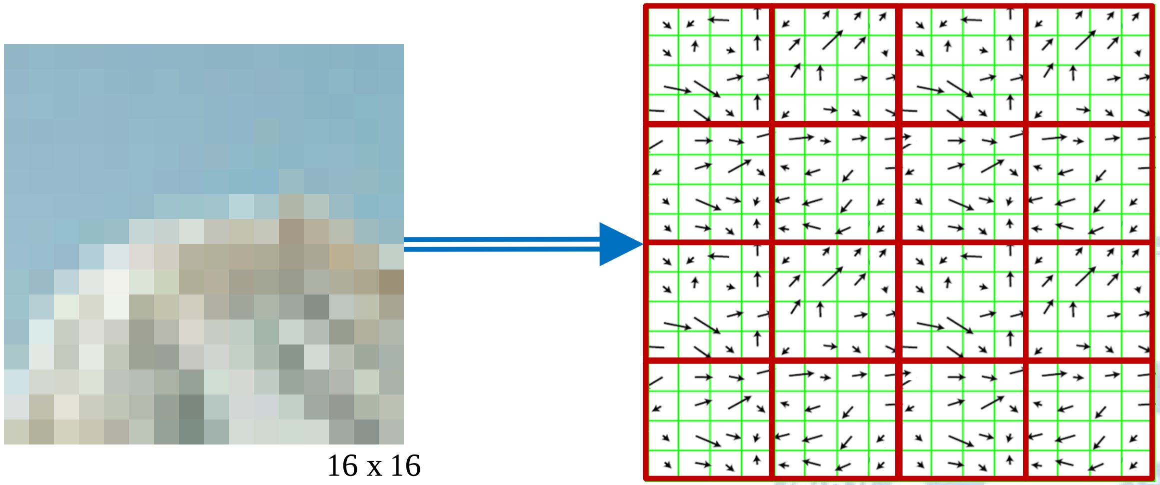
How to compute SIFT descriptor?

Input: an image and a location to compute the descriptor

Step 1: Warp the image to the correct orientation and scale, and than extract the feature as 16×16 pixels

Descriptors – SIFT (A brief introduction)

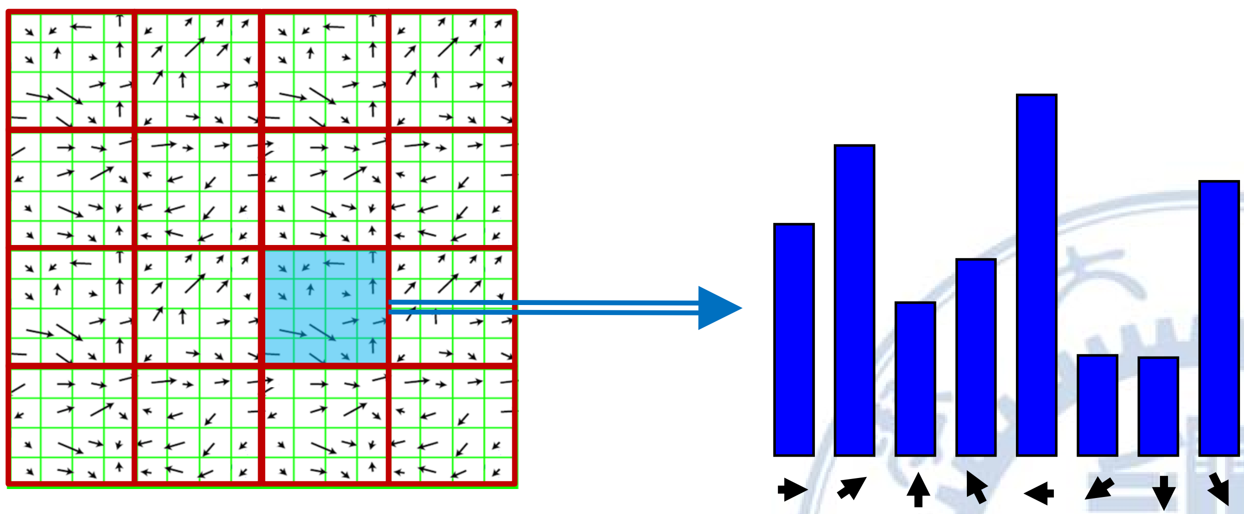
Step 2: Compute the gradient for each pixel (direction and magnitude)



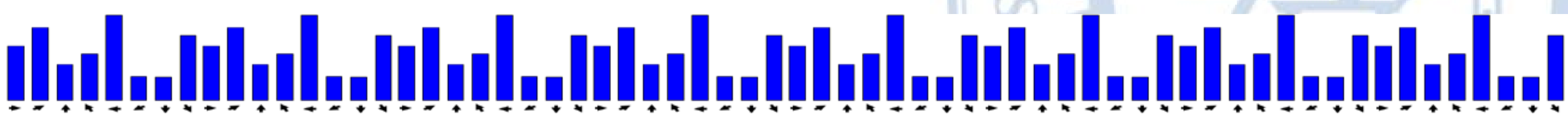
Step 3: Divide the pixels into 4×4 squares

Descriptors – SIFT (A brief introduction)

Step 4: For each square, compute gradient direction histogram over 8 directions. Choose the peak of that histogram as Keypoint orientation and rotate all of the directions so that Keypoint orientation=0^0.

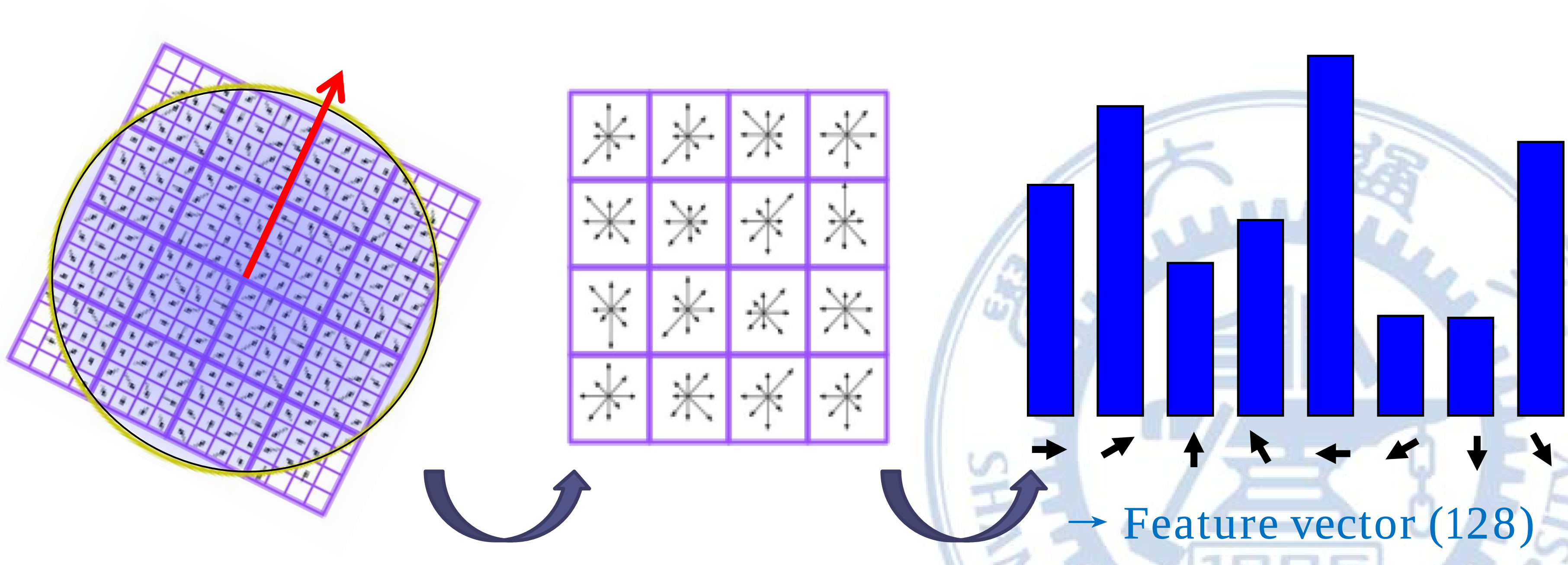


The result: 128 dimensions feature vector.



Conclusion

- Warp the feature into 16×16 square.
- Divide into 4×4 squares.
- For each square, compute an histogram of the gradient directions.



Conclusion

Invariance to illumination

- Gradient are invariant to Light intensity shift (i.e. add a scalar to all the pixels)
- Normalization to unit length add invariance to light intensity change (i.e. multiply all the pixels by a scalar)

Invariance to shift and rotation

- Histograms does not contains any geometric information
- Using 16 histograms allows to preserve geometric information

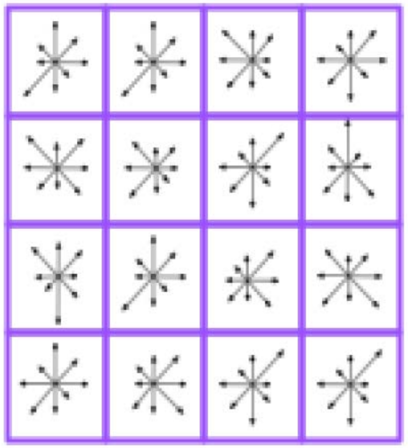


Image Descriptors

- Chose a representation scheme
  - Chain codes (4- and 8-directional chain codes)
  - Polygonal approximation
  - Signature (1-D functional representation)
- Describe the region based on the scheme
  - Boundary descriptors
    - Length, diameter, shape numbers, Fourier descriptors
  - Region descriptors
    - Area, compactness, principal axes
    - Moment , Texture

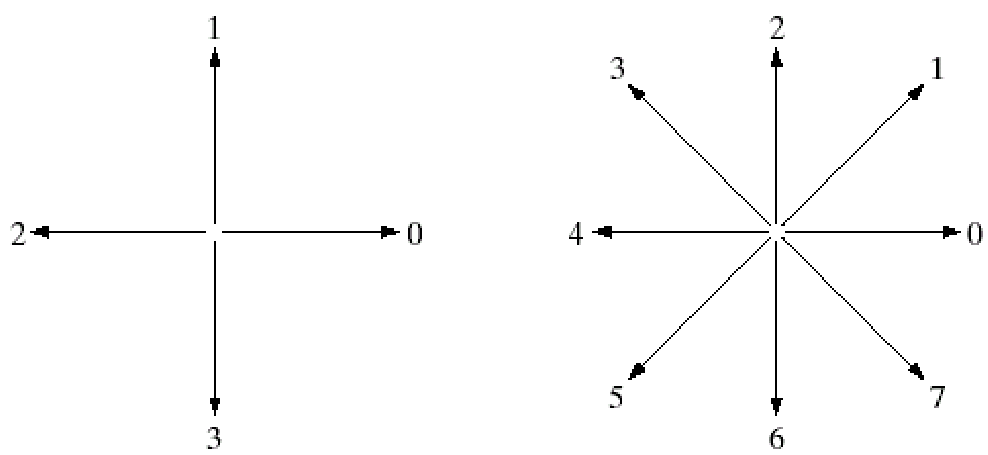
- It is important that the features selected as descriptors be as insensitive as possible to such variations as changes in scale, translation, and rotation

Chain codes

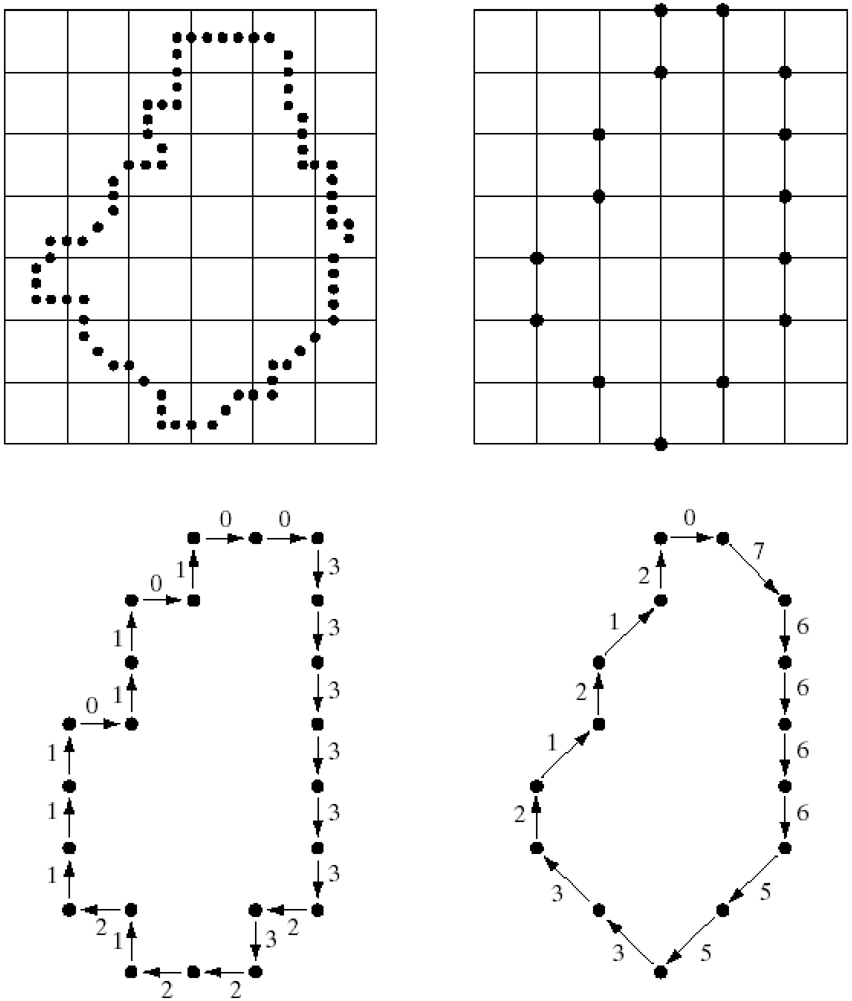
- Represent a boundary by a connected sequence of straight-line segments of specified length and direction
- 4-directional chain codes
- 8-directional chain codes

a b

FIGURE 11.1 Direction numbers for (a) 4-directional chain code, and (b) 8-directional chain code.



Chain codes - example



a b
c d

FIGURE 11.2 (a) Digital boundary with resampling grid superimposed. (b) Result of resampling. (c) 4-directional chain code. (d) 8-directional chain code.

Chain codes normalization

- Problem 1: different starting points result in different chain codes
  - normalization → redefine the starting point such that the chain code forms a smallest number

code = {3,4,3,4,4,5,4,6,7,7,7,0,0,1,1,2} (a) Initial chain code	code = {4,3,4,4,5,4,6,7,7,7,0,0,1,1,2,3} (b) Result of one shift
code = {3,4,4,5,4,6,7,7,7,0,0,1,1,2,3,4} (c) Result of two shifts	code = {0,0,1,1,2,3,4,3,4,4,5,4,6,7,7,7} (d) Minimum integer chain code

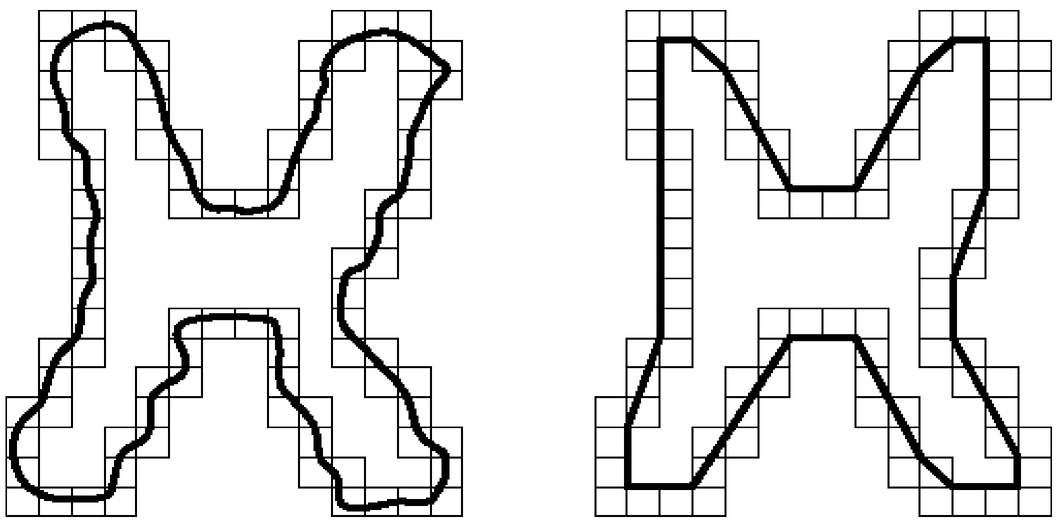
- Problem 2: object rotation results in different chain codes
  - difference code → coding with the difference of directions (counter-clockwise)
  - E.g.: 0000655332 → 0006706076 → 0006706076

Polygonal approximations

- Approximate a boundary using a polygon
  - Minimum perimeter polygons
  - Merging techniques
  - Splitting techniques

Minimum perimeter polygons

- Choose an appropriate grid → The boundary is enclosed by a set of concatenated cells
- Allow the boundary to shrink as a rubber band

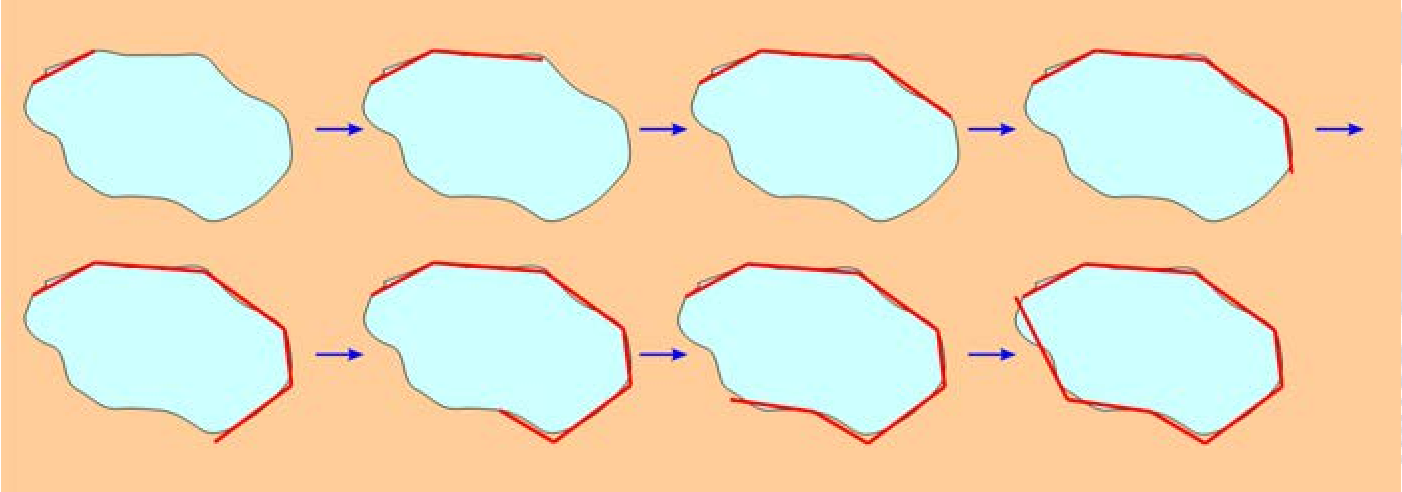


a b

FIGURE 11.3 (a) Object boundary enclosed by cells. (b) Minimum perimeter polygon.

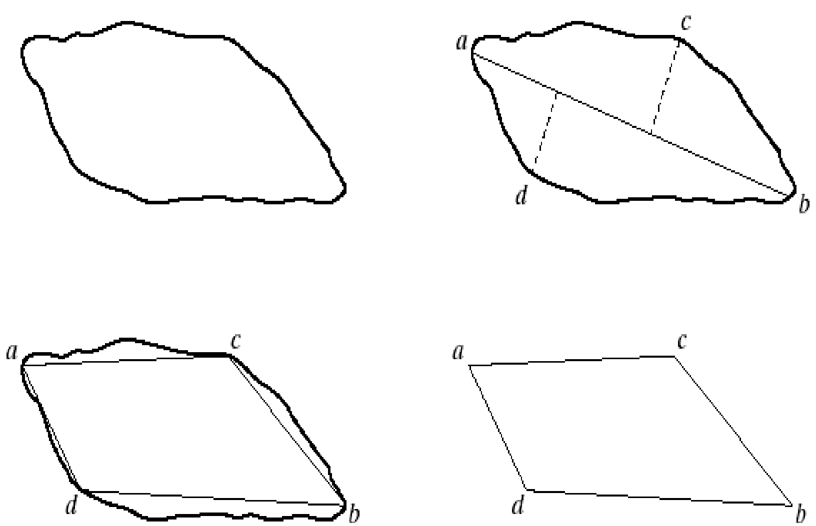
Merging techniques

- Step 1:
  - Merge points along a boundary until the least square error line fit of the points merged so far exceeds a threshold
- Step 2:
  - Record the two end point of the line
- Step 3:
  - Repeat Steps 1 and 2 until all boundary points are processed



Splitting technique

- Successively subdivide a segment into two parts until a given criterion is satisfied.
- One can use the major axis as the first subdivided line.
- The stopping criterion can be taken as 0.25 times the length of the major axis



a b
c d

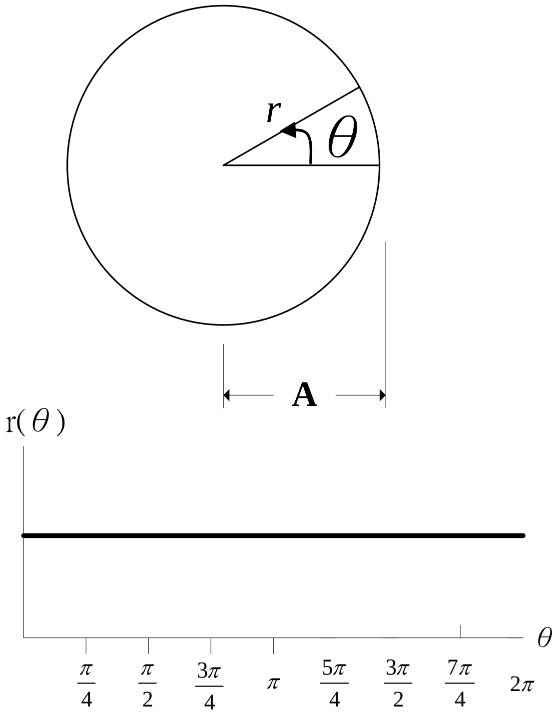
FIGURE 11.4
(a) Original boundary.
(b) Boundary divided into segments based on extreme points. (c) Joining of vertices.
(d) Resulting polygon.

Signature

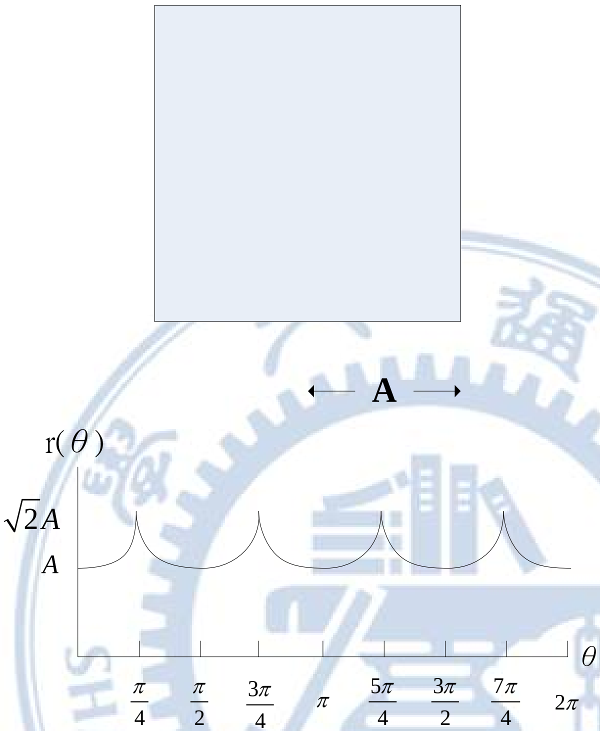
- A 1-D functional representation of a boundary
- Basic idea : reduce the boundary representation to a 1-D function, which might be easier to describe than a 2-D boundary
- One simple approach: use the distance from the centroid to the boundary as a function of angle. It is invariant to translation, but not to rotation and scaling.
  - Rotation : select the farthest point from the centroid as the starting point
  - Scaling : normalize the function by variance

Signature

- Distance signature of circle shapes



- Distance signature of rectangular



Boundary descriptors

- Length
  - For a chain-coded curve with unit spacing
  - Length = the number of vertical and horizontal components + the number of diagonal components $\times$ $2^{1/2}$
- Diameter
  - Maximum distance between any two points on the boundary
  - The line formed by this two points is called the major axis of the boundary
- Fourier descriptors

Fourier descriptors

- Points on the boundary can be treated as a complex number

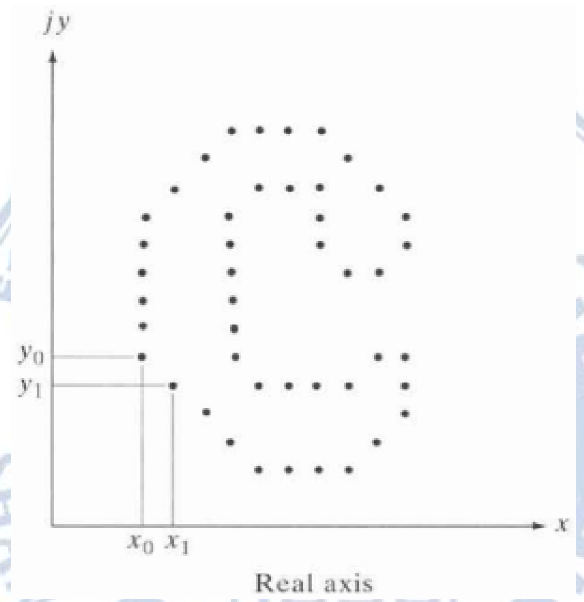
$$s(k) = x(k) + jy(k)$$

- Fourier descriptor : the discrete Fourier transform (DFT) of the $s(k)$

$$a(u) = \frac{1}{N} \sum_{k=0}^{N-1} s(k) \exp[-j2\pi \cdot uk/N]$$

- Reconstruction: if $a(u) = 0$, for $u > N-1$

$$\hat{s}(k) = \sum_{u=0}^{N-1} a(u) \exp[j2\pi \cdot uk/N]$$



Fourier descriptors

- Usually, only the first few coefficients are used to represent the shape
- Disadvantage:
  - Just for closed boundaries

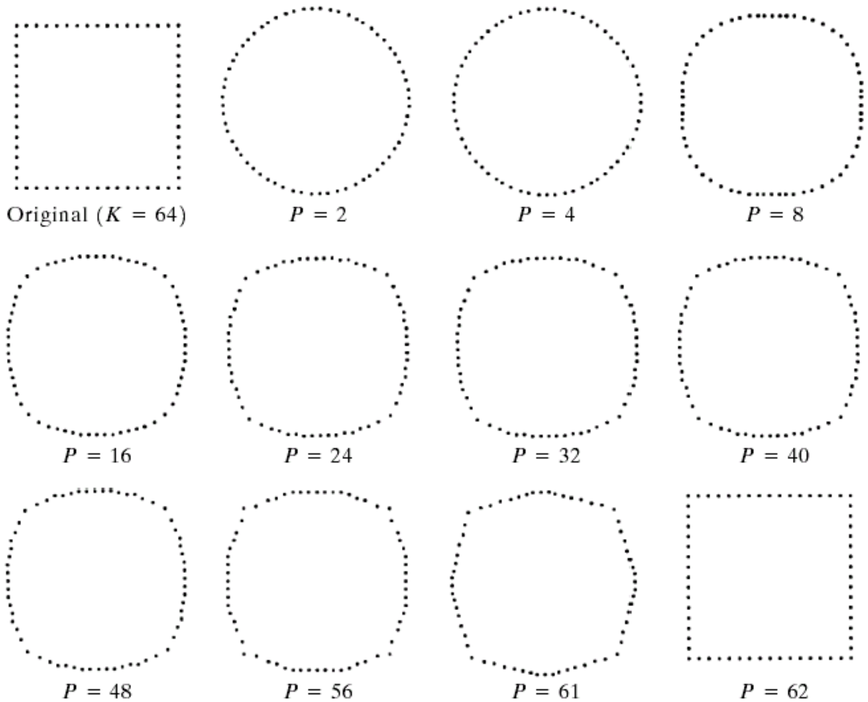


FIGURE 11.14 Examples of reconstruction from Fourier descriptors. P is the number of Fourier coefficients used in the reconstruction of the boundary.

Simple region descriptor

- Area
  - The number of pixels contained within its boundary

$$A(S) = \sum_{x,y} 1 \qquad (x, y) \in S$$

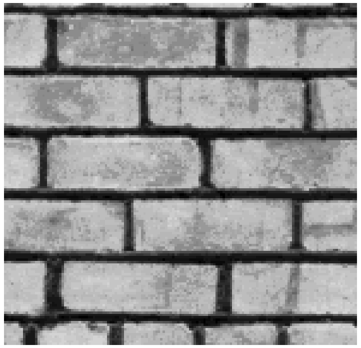
- Perimeter
  - The length of its boundary
  - x(t) and y(t) denote the parametric co-ordinates of a curve enclosing a region S

$$P(S) = \int_t \sqrt{x^2(t) + y^2(t)}dt$$

$$P(S) = \sum_i \sqrt{(x_i - x_{i-1})^2 + (y_i - y_{i-1})^2}$$

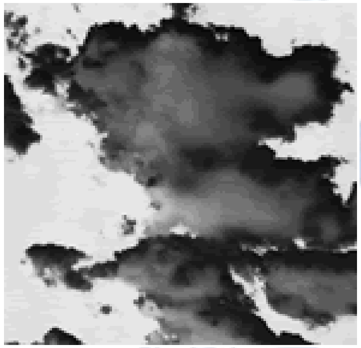
Texture

- No formal definition of texture exists
  - repeating patterns of local variations in image intensity, which is too fine to be distinguished
  - relates mostly to a specific, spatially repetitive (micro) structure of surfaces formed by repeating a particular element or several elements in different relative spatial positions.

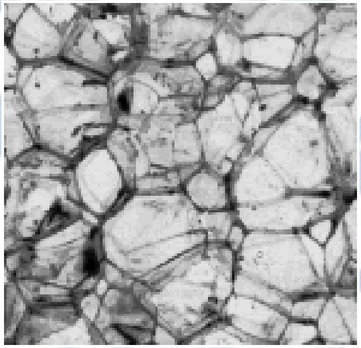


Brick Texture

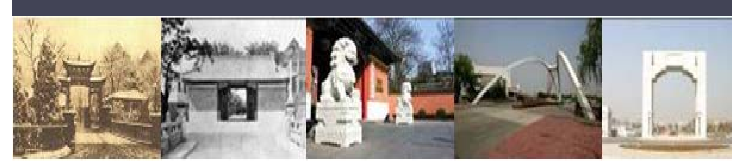
Finger print Texture



Clouds Texture



Rocks Texture



Thank You!